MindSpark – Quiz App

Author

Name: Mohd Zakie Khan

Roll: 22f1001965

Email: 22f1001965@ds.study.iitm.ac.in

About Me: I'm a fresher currently in diploma level of BS in Data Science and its application. My aim is to become a skilled data scientist and well as a web developer. Currently having hand on experience on React JS, Vue JS and Flask.

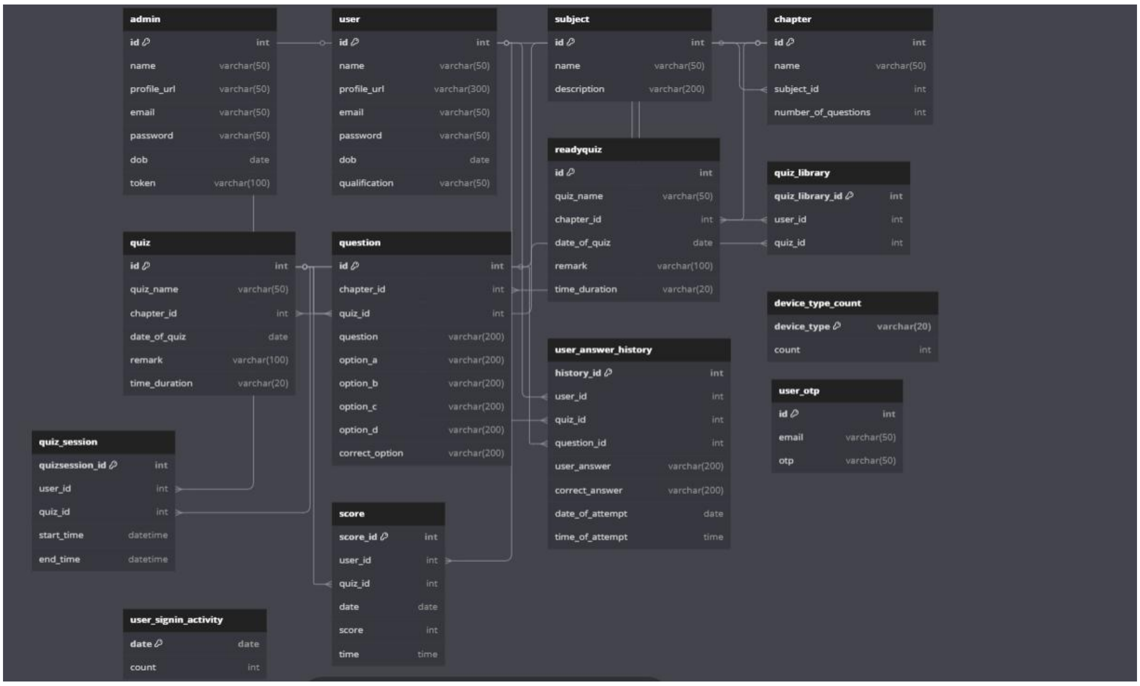
Project Description

This project involves making of an app where user can attempt a quiz based upon their preference and can also evaluate their performance on their user dashboard. Admin can edit/delete/create new quizzes and can delete user based upon his or her need.

Technologies used

- Vue3 – A progressive JavaScript framework for building interactive user interfaces.
- Flask – A lightweight Python web framework for building web applications and APIs.
- Bootstrap – A popular CSS framework for responsive and mobile-first web design.
- Axios – A JavaScript library for making HTTP requests, often used with Vue.
- Flask SQL Alchemy – A Flask extension for integrating SQL Alchemy, an ORM for database management.
- Celery – A distributed task queue for handling asynchronous background tasks.
- Redis – An in-memory data store used for caching, message brokering, and real-time analytics.
- JWT (JSON Web Token) – A secure method for transmitting authentication and authorization data.
- Flask-Mail – A Flask extension for sending emails using SMTP.

DB Schema



Architecture and Features

The Repository consist of 2 folder- (a)For Vue JS (b) For Flask Server

a) Flask Server Folder:

Controller: Which consist of User Controller and Admi Controller respectively

Model: Consist of Database Schema with the celery configs and its task

Instance Folder: Contain the quiz dB file

b) Vue Folder:

Src Folder: Consist Various Component of Web-App

a. Assests: Contains CSS, JS and images

b. Components: Contains Reusable Web Components

c. View: Contain Layout of Web Pages

d. Router: Contains the URL for difference page in web app

Root Folder: Contains index.js

Feature Implementation

- **User Authentication** – Users and admins can register, log in, and authenticate securely.
- **Quiz Management** – Admins can create, update, and manage quizzes with chapters and subjects.
- **Question Bank** – Stores and manages multiple-choice questions with correct answers.
- **Quiz Participation** – Users can take quizzes with a set duration and track progress.
- **Scoring System** – Calculates and records user scores upon quiz completion.
- **Answer Tracking** – Stores user responses and compares them with correct answers.
- **Quiz History** – Maintains past quiz attempts for review and analysis.
- **OTP Verification** – Secure login and authentication via OTP-based verification.
- **Predefined Quizzes** – Ready-made quizzes available for quick participation.
- **Performance Analytics** – Tracks quiz participation, scores, and activity insights.
- **Time-Tracked Sessions** – Logs quiz start and end times for session tracking.
- **Admin Dashboard** – Admins can monitor quiz performance, users, and activity.
- **Email Notifications** – Users receive emails for authentication and quiz updates.

API Endpoints:

Admin:

- /admin/create – Creates a new admin account.
- /admin/authorize – Authenticates an admin user.
- /admin/getinfo – Fetches general admin-related information.
- /admin/get_subject – Retrieves subject details from the database.
- /admin/add_subject – Adds a new subject to the system.
- /admin/delete_subject – Deletes an existing subject.
- /admin/add_chapter – Adds a new chapter under a subject.
- /admin/get_chapter – Retrieves chapter details.
- /admin/edit_chapter – Updates an existing chapter’s details.
- /admin/delete_chapter – Removes a chapter from the system.
- /admin/add_quiz – Creates a new quiz under a chapter.
- /admin/get_quizes – Fetches a list of all quizzes.
- /admin/get_ready_quizes – Retrieves quizzes that are marked as “ready” for users.
- /admin/add_ready_quiz – Marks a quiz as “ready” for publishing.
- /admin/remove_ready_quiz – Unmarks a quiz from the ready list.
- /admin/add_question – Adds a new question to a quiz.
- /admin/get_questions – Retrieves all questions for a quiz.
- /admin/get_question – Fetches details of a specific question.
- /admin/delete_question – Deletes a question from a quiz.
- /admin/edit_question – Updates a quiz question.

- /admin/get_quiz – Retrieves details of a specific quiz.
- /admin/search – Allows searching for quizzes, questions, or users.
- /admin/delete_user – Removes a user from the system.
- /admin/get_user_data – Fetches user details for the admin panel.

User:

- /user/create_user – Registers a new user.
- /user/user_authorize – Logs in a user.
- /user/add_device – Associates a device with a user account.
- /user/get_profile_data – Fetches a user's profile information.
- /user/get_quizes – Retrieves available quizzes for the user.
- /user/get_ready_quizes – Fetches quizzes that are ready to be taken.
- /user/add_to_library – Saves a quiz to the user's personal library.
- /user/get_my_quizes – Lists quizzes saved in the user's library.
- /user/check_quiz_in_library – Checks if a quiz is in the user's library.
- /user/get_questions – Fetches questions for a quiz the user is taking.
- /user/start_quiz – Begins a quiz session.
- /user/remaining_time – Gets the remaining time for an ongoing quiz.
- /user/submit_quiz – Submits answers for a completed quiz.
- /user/get_score – Retrieves the user's score after submitting a quiz.
- /user/get_pdf – Generates and downloads a PDF report of quiz results.
- /user/get_user_chart – Retrieves graphical performance data for the user.
- /user/search – Allows searching for quizzes or user data.
- /user/save_img – Uploads and saves an image (possibly a profile picture).
- /user/get_radar_chart – Fetches a radar chart representing user performance.
- /user/get_user_answer – Retrieves the user's answers for a quiz.
- /user/get_user_data – Gets the user's personal data.
- /user/edit_profile – Updates user profile details.
- /user/get_stats – Retrieves statistics about the user's quiz performance.

Celery:

- /celery/get_csv_data – Retrieves data in CSV format.
- /celery/send_report – Sends reports via Celery task processing.

Demonstration Video:

URL:
https://drive.google.com/file/d/1dytEzDPtNc302VgKJCvzLaT5YBIE3T24/view?usp=sharing